



Systeme de configuration ispFlash

O. Bourrion

► To cite this version:

| O. Bourrion. Systeme de configuration ispFlash. 2003. in2p3-00014000

HAL Id: in2p3-00014000

<https://hal.in2p3.fr/in2p3-00014000>

Preprint submitted on 17 Sep 2003

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Système de configuration ispFlash

Table des Matières :

1	<u>INTRODUCTION</u>	2
2	<u>BROCHAGE DU PORT PARALLÈLE</u>	2
3	<u>EPP TIMING</u>	3
4	<u>INTERCONNEXION DE LA MÉMOIRE ET DE L'EPLD</u>	5
5	<u>DESCRIPTION FONCTIONNELLE DE L'EPLD</u>	6
5.1	SYNOPTIQUE	6
5.2	DESCRIPTION	7
5.3	MODULE EPP_INTERFACE	7
5.3.1	INTERFACE	7
5.3.2	SYNOPTIQUE	8
5.4	MODULE SERIALIZER	8
5.4.1	INTERFACE	8
5.4.2	SYNOPTIQUE	9
6	<u>SOFTWARE DE CONTRÔLE</u>	10
6.1	FENÊTRE DE DESCRIPTION D'UNE FLASH	10
6.2	FENÊTRE PRINCIPALE	11

Table des figures :

<i>Figure 1 : Cycle de lecture d'adresse EPP</i>	3
<i>Figure 2 : Cycle d'écriture d'adresse EPP</i>	3
<i>Figure 3 : Cycle de lecture de donnée EPP</i>	4
<i>Figure 4 : Cycle d'écriture de donnée EPP</i>	4
<i>Figure 5 : schéma du module ispFlash</i>	5
<i>Figure 6 : Synoptique de l'EPLD ispFlash</i>	6
<i>Figure 7 : Interface du module EPP</i>	7
<i>Figure 8 : FSM de EPP interface</i>	8
<i>Figure 9 : interface du module serializer</i>	8
<i>Figure 10 : FSM de programmation du FPGA</i>	9

Liste des tableaux :

<i>Tableau 1 : Brochage du port parallèle</i>	2
---	---

ANNEXE :

Vue du module

Schéma du module

1 Introduction

Le but de ce module est de permettre d'utiliser des composants FPGA à la place des EPLD qui sont plus onéreux à capacité égale ou moindre. Par exemple, l'idée est de remplacer des composants CPLD ayant 512 macrocellules par un FPGA spartan II de Xilinx. Cependant à cause de la volatilité de la configuration, il faut avoir un moyen de configuration à la mise sous tension. La solution est d'utiliser une mémoire Flash de forte capacité couplée à un CPLD de faible taille pour prendre en charge le protocole de configuration du FPGA ; Il faut aussi prévoir le moyen de configurer cette mémoire insitu. Bien entendu, un produit présentant une fonctionnalité équivalente existe déjà, puisque Xilinx et ALTERA fournissent des PROMs de configuration sérielle. Malheureusement elles sont chères et il faut passer par un distributeur alors que la FLASH, le petit CPLD et les FPGA spartan II sont disponibles couramment.

Bref, en utilisant ce montage, au prix d'un peu de surface supplémentaire et d'un délai de configuration maximum (pour le plus gros FPGA 1Mbit) de 240 ms, on obtient une solution moins chère et plus performante qu'un composant EPLD de forte capacité.

2 Brochage du port parallèle

Pin	SPP Signal	EPP Signal	IN/OUT	Function
1	Strobe	Write	Out	A low on this line indicates a Write, High indicates a Read
2-9	Data 0-7	Data 0-7	In-Out	Data Bus. Bi-directional
10	Ack	Interrupt	In	Interrupt Line. Interrupt occurs on Positive (Rising) Edge.
11	Busy	Wait	In	Used for handshaking. A EPP cycle can be started when low, and finished when high.
12	Paper Out / End	Spare	In	Spare - Not Used in EPP Handshake
13	Select	Spare	In	Spare - Not Used in EPP Handshake
14	Auto Linefeed	Data Strobe	Out	When Low, indicates Data transfer
15	Error / Fault	Spare	In	Spare - Note used in EPP Handshake
16	Initialize	Reset	Out	Reset - Active Low
17	Select Printer	Address Strobe	Out	When low, indicates Address transfer
18-25	Ground	Ground	GND	Ground

Tableau 1 : Brochage du port parallèle

3 EPP timing

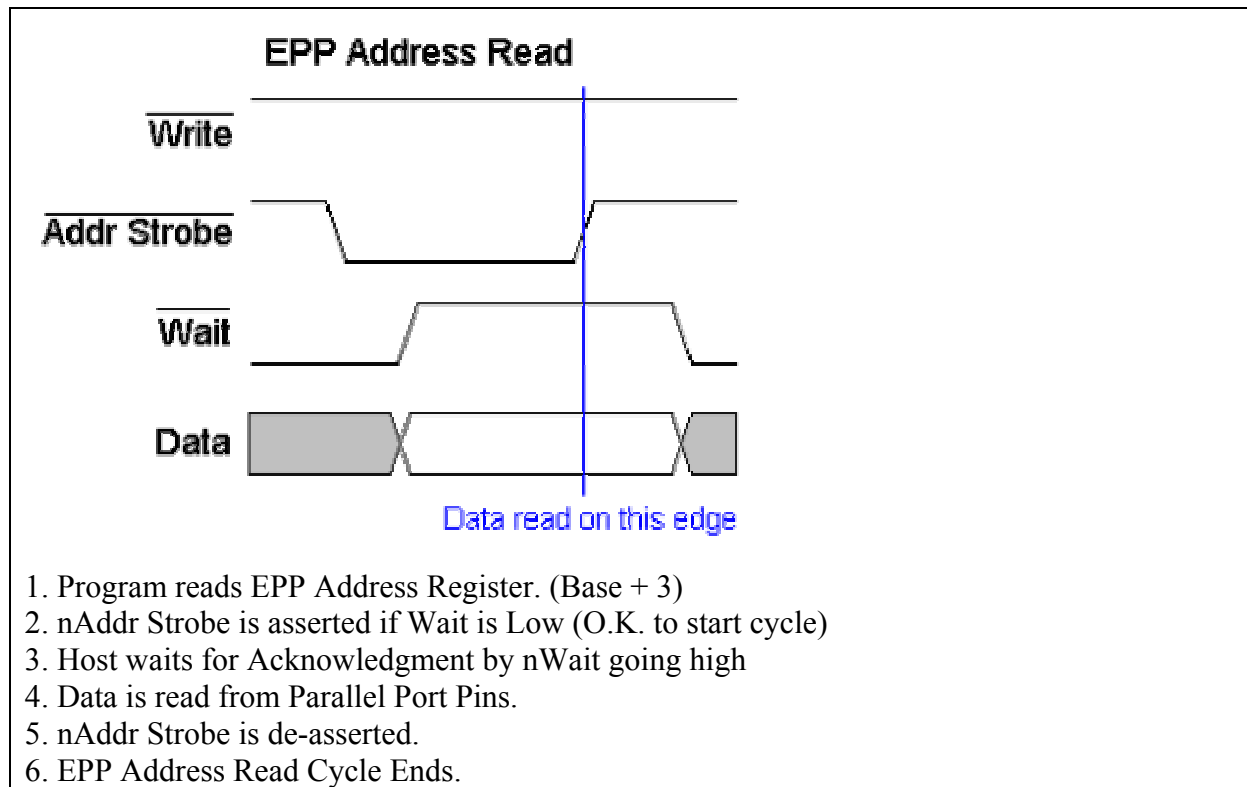


Figure 1 : Cycle de lecture d'adresse EPP

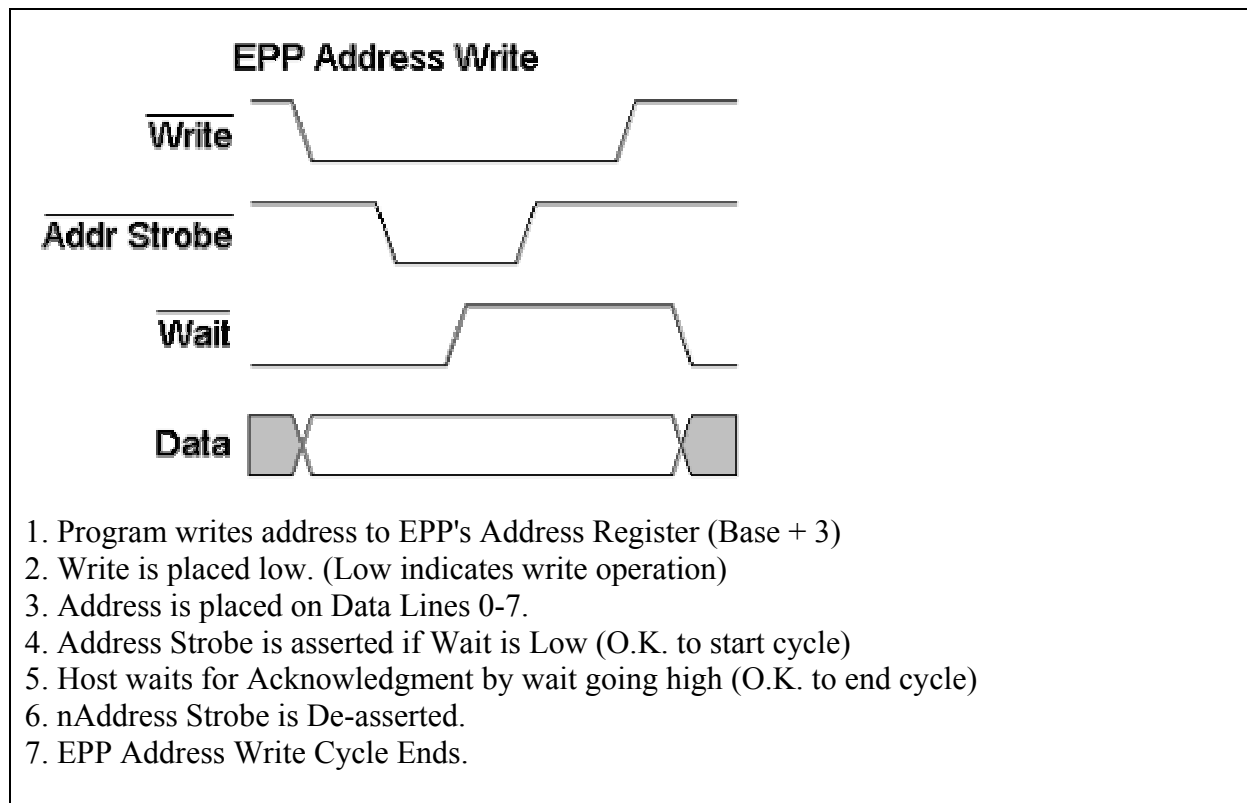


Figure 2 : Cycle d'écriture d'adresse EPP

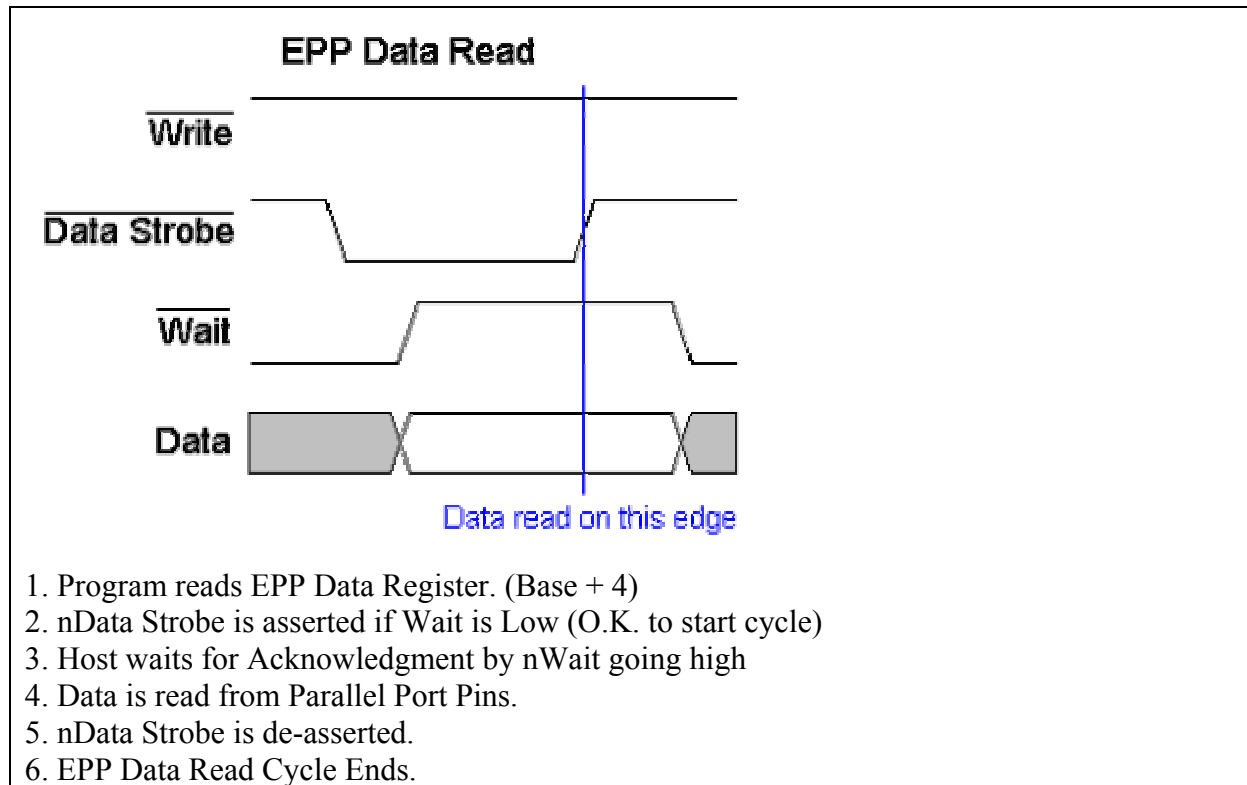


Figure 3 : Cycle de lecture de donnée EPP

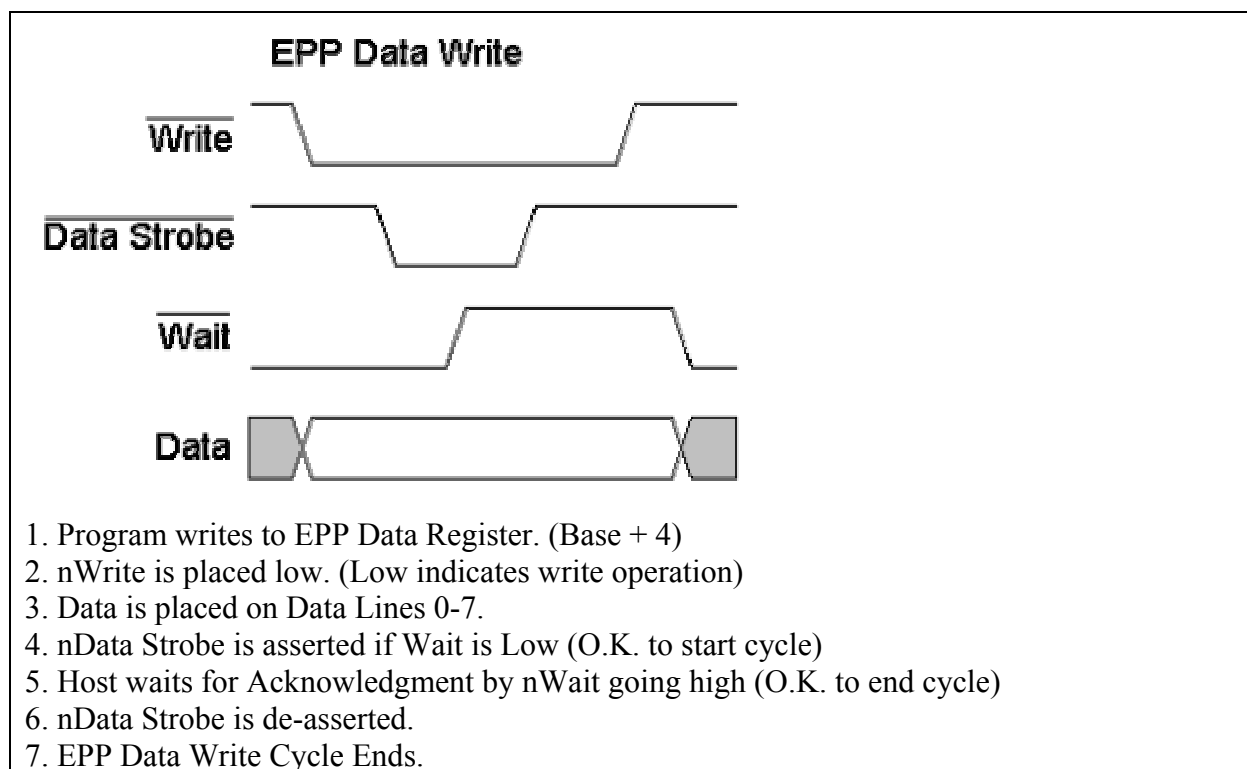


Figure 4 : Cycle d'écriture de donnée EPP

4 Interconnexion de la mémoire et de l'EPLD

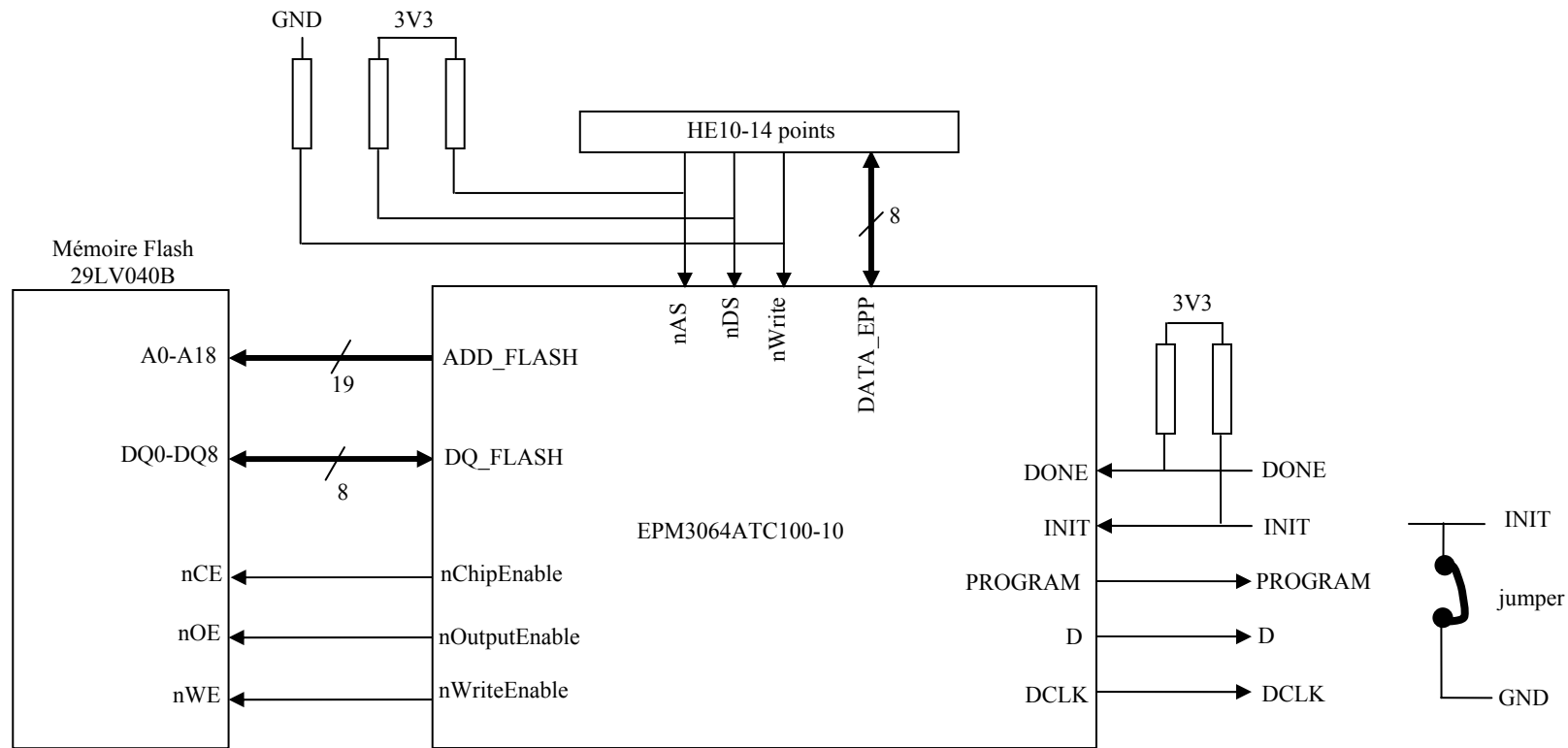


Figure 5 : schéma du module ispFlash

5 Description fonctionnelle de l'EPLD

5.1 Synoptique

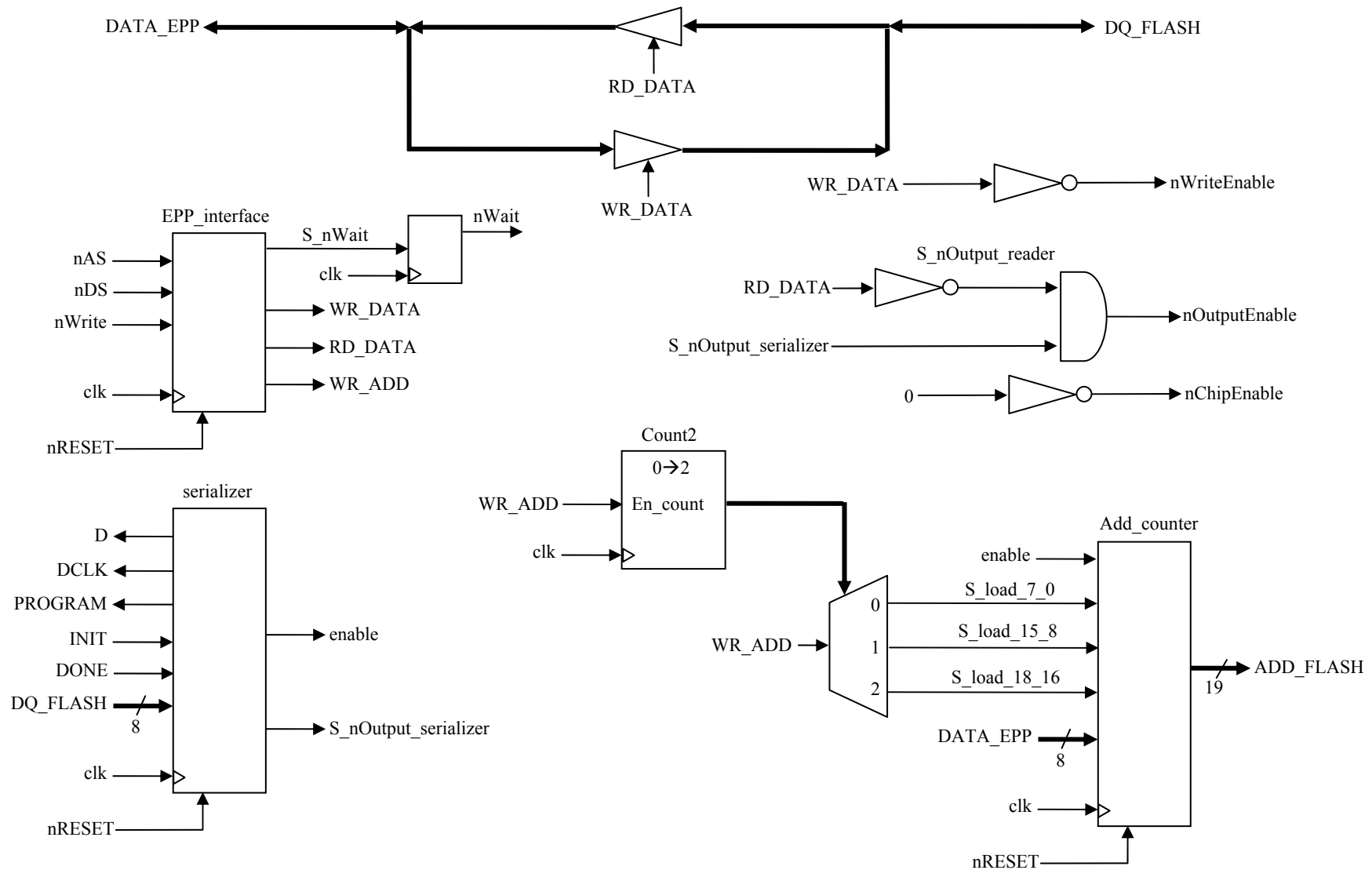


Figure 6 : Synoptique de l'EPLD ispFlash

5.2 Description

Le module EPP interface sert à prendre en charge le protocole du port parallèle EPP. En sortie il fournit des signaux synchronisés sur l'horloge du bus VME.

Le module serializer est actif tant que le FPGA n'est pas configuré ou en erreur de configuration. Lors de la première configuration de la FLASH ou après un effacement, le FPGA ne se mettant pas en erreur de configuration, il faut forcer le signal init grâce à un jumper. Ainsi l'accès à la Flash est possible.

Lors de la configuration de la mémoire (accès en écriture), il est nécessaire de faire un cycle d'écriture d'adresse pour prépositionner l'adresse désirée. Cela est fait grâce au fait que le compteur d'adresse est prechargeable. Pour ce faire, il faut faire trois écritures successives.

Attention, l'accès direct à la mémoire par l'EPP ne peut fonctionner qu'une fois le FPGA configuré ou en erreur de configuration.

5.3 Module EPP_interface

5.3.1 Interface

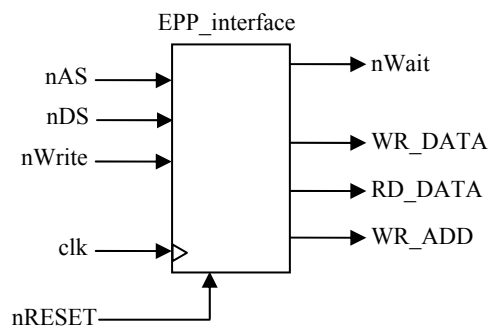


Figure 7 : Interface du module EPP

- **nAS** : Address Strobe, signal asynchrone indiquant un cycle d'adresse EPP ;
- **nDS** : Data Strobe, signal asynchrone indiquant un cycle de données EPP ;
- **nWrite** : Signal asynchrone indiquant le sens de la transaction sur le bus EPP (0=écriture) ;
- **nWait** : signal de handshake indiquant que le terminal est prêt à compléter l'opération demandée ;
- **Wr_data** : signal synchrone, d'une durée d'un cycle d'horloge indiquant la présence de donnée valide sur le bus EPP ;
- **RD_DATA** : signal synchrone durant aussi longtemps que le cycle de donnée est actif (nDS=0) indiquant une demande lecture ;
- **WR_ADD** : signal synchrone, d'une durée d'un cycle d'horloge indiquant la présence d'une adresse valide sur le bus EPP ;
- **clk** : horloge cadencée à 16 MHz ;
- **nRESET** : signal asynchrone de reset ;

- **D** : signal de sortie donnée série à destination du FPGA ;
- **DCLK** : signal de cadencement des données séries ;
- **PROGRAM** : signal de mise en configuration du FPGA ;
- **INIT** : signal indiquant la fin de l'effacement du FPGA ou une erreur en cours de programmation ;
- **DONE** : signal indiquant la fin de la configuration du FPGA ;
- **Enable** : impulsion d'un cycle d'horloge permettant d'incrémenter le compteur d'adresse ;
- **S_nOutput_serializer** : signal permettant de mettre la mémoire en mode lecture ;

5.4.2 Synoptique

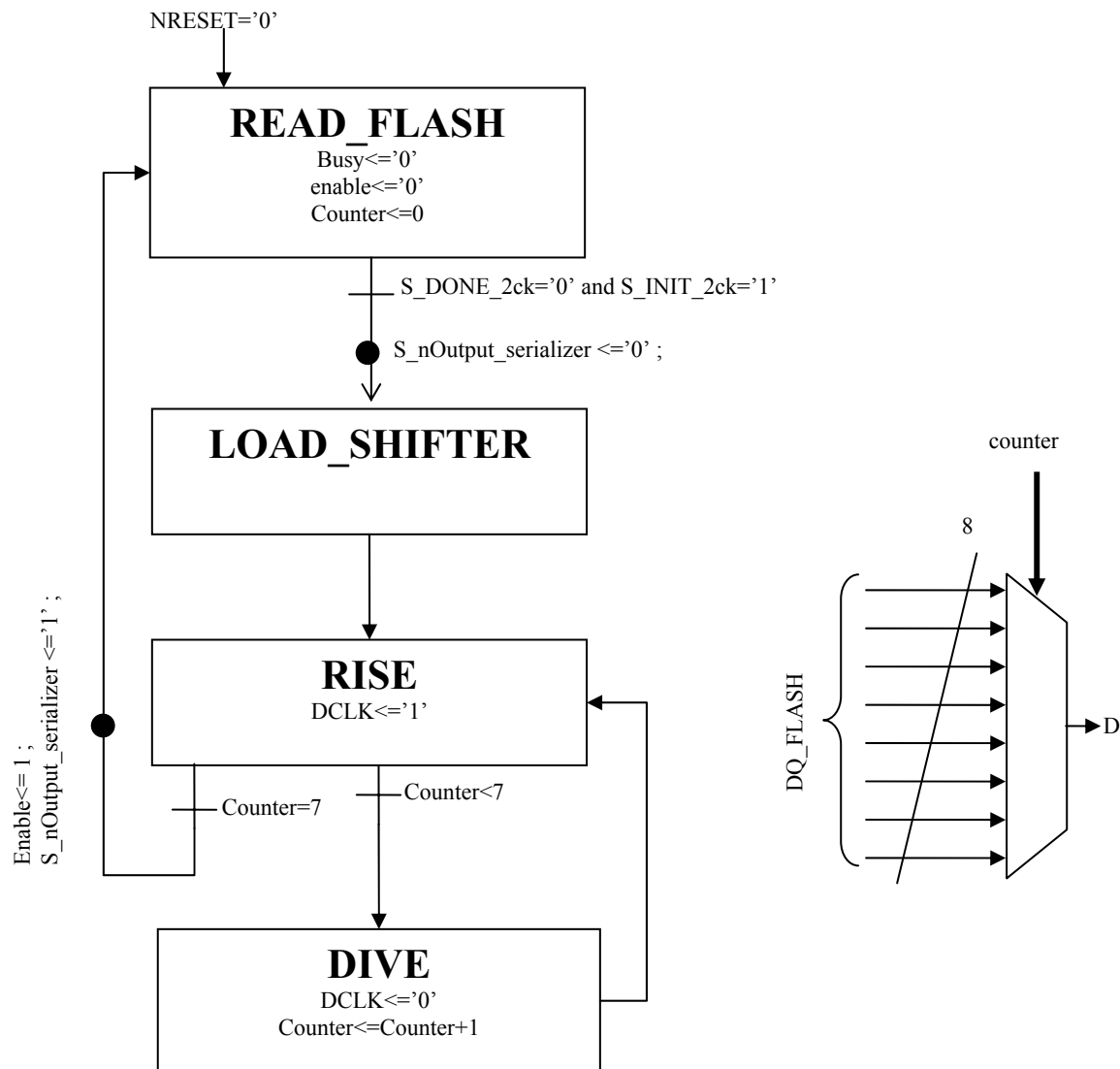


Figure 10 : FSM de programmation du FPGA

Les signaux **nDS** et **nAS** sont resynchronisés en interne pour éviter toute métastabilité. **nWrite** étant stable une fois que **nDS** ou **nAS** est activé.

6 Software de contrôle

Un soft a été conçu pour pouvoir communiquer à travers le port parallèle et pour pouvoir décrire les différents modèles de Flash.

La description du soft est ici très sommaire, car il est documenté à travers les fichier d'aide windows.

6.1 Fenêtre de description d'une Flash

The screenshot shows a window titled "FlashBuilder - C:\Am29LV040B.flash". It contains several input fields and checkboxes for configuring a flash device. The parameters are as follows:

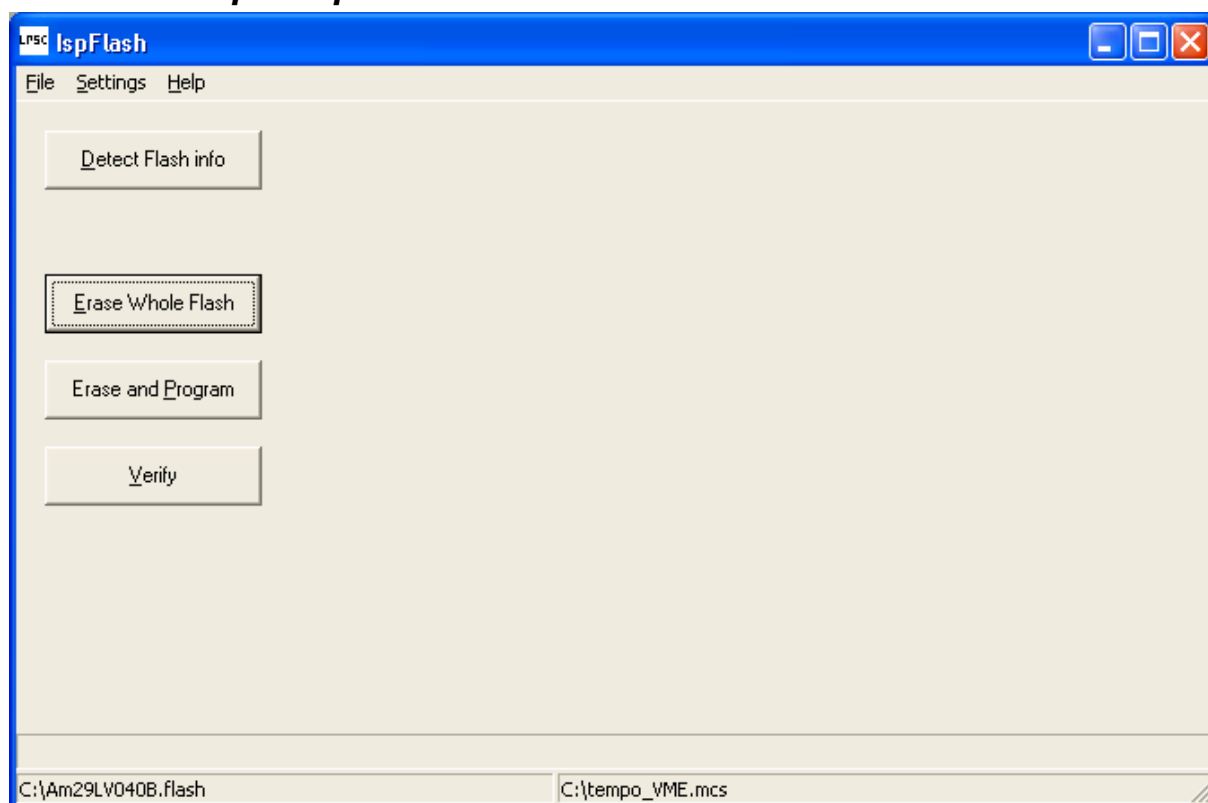
Block/sector number	offset address
0	0
1	10000
2	20000
3	30000
4	40000
5	50000
6	60000
7	70000

The window also includes a menu bar with "File" and "Help", and a button labeled "Erase Block/Sector Address Table".

Cette fenêtre donne tous les paramètres nécessaires pour pouvoir communiquer avec une Flash :

- Manufacturer ID
- Device ID
- La taille en puissance de 2
- Les adresses ou envoyer les codes commande
- La disponibilité d'une commande bypass permettant de doubler la vitesse de programmation

6.2 Fenêtre principale



Une fois le modèle de flash sélectionné et le fichier de programmation chargé en mémoire, le soft est prêt à communiquer avec la Flash.

ISPFLASH

